

MM MM SSSSSSSS GGGGGGGG DDDDDDDD EEEEEEEEEE FFFFFFFFFF
MM MM SSSSSSSS GGGGGGGG DDDDDDDD EEEEEEEEEE FFFFFFFFFF
MMMM MMMM SS GG DD DD EEE FF
MMMM MMMM SS GG DD DD EEE FF
MM MM MM SS GG DD DD EEE FF
MM MM SS SSSSSS GG DD DD EEE FF
MM MM SSSSSS GG DD DD EEE FF
MM MM SS SSSSSS GG DD DD EEE FF
MM MM SS GG GGGGGG DD DD EEE FF
MM MM SS GG GGGGGG DD DD EEE FF
MM MM SS GG GG DD DD EEE FF
MM MM SS GG GGGGGG DDDDDDDD EEEEEEEEEE FF
MM MM SSSSSSSS GGGGGG DDDDDDDD EEEEEEEEEE FF

....
....
....

SSSSSSSS DDDDDDDD LL
SSSSSSSS DDDDDDDD LL
SS DD DD LL
SSSSSSSS DDDDDDDD LLLLLLLL
SSSSSSSS DDDDDDDD LLLLLLLL

{-----

Structure definitions for message sections

Version 'V04-000'

```
*****  
*  
* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
* ALL RIGHTS RESERVED.  
*  
* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
* TRANSFERRED.  
*  
* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
* CORPORATION.  
*  
* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
*  
*****
```

{
{
{

Define the message section format

module \$MSCDEF;

```
aggregate MSCDEF structure prefix MSCS;  
    TYPE byte unsigned; /* TYPE OF MESSAGE SECTION  
    constant(  
        MSG  
        , IND  
        , MAXTYPE  
    ) equals 0 increment 1 prefix MSC tag SC;  
    FLAGS OVERLAY union;  
        FLAGS byte unsigned; /* FLAGS  
        FLAGS BITS structure;  
            MAPPED bitfield mask;  
        end FLAGS BITS;  
    end FLAGS OVERLAY;  
    SANITY word unsigned; /* SANITY CHECK (MUST = MSC$SC_SANITY)  
    constant SANITY equals 1234 prefix MSC tag SC;  
    SIZE longword unsigned; /* LENGTH OF ENTIRE MESSAGE SECTION (MSG)  
    INDEX_OFF longword unsigned; /* OFFSET TO PRIMARY INDEX PORTION
```

```

FAC OFF longword unsigned;           /* OFFSET TO FACILITY TABLE
TEXT OFF longword unsigned;         /* OFFSET TO TEXT PORTION
SUBST OFF longword unsigned;        /* OFFSET TO TEXT SUBSTITUTION TABLE
FILL_T longword dimension 4 fill prefix MSCDEF tag $$; /* RESERVED
constant "LENGTH" equals . prefix MSC$ tag K;          /* LENGTH OF FIXED PORTION (MSG)
constant "LENGTH" equals . prefix MSC$ tag C;          /* LENGTH OF FIXED PORTION (MSG)

end MSCDEF;

aggregate MSCDEF1 structure prefix MSC$;
  FILL 2 byte dimension 8 fill prefix MSCDEF tag $$;
  INDNAMELEN byte unsigned;           /* LENGTH OF COUNTED STRING
  INDNAME character;                /* FILE SPEC OF MESSAGE FILE
end MSCDEF1;

end_module $MSCDEF;

module $MIDXDEF;

/*
/* Define message section index
/*
/* All message indicies are longword aligned and the
/* size is always rounded to the next longword boundary.
/* This is done for additional verification checks.
*/

constant BKTSIZ equals 512 prefix MIDX tag $C;           /* BUCKET SIZE IN BYTES

aggregate MIDXDEF structure prefix MIDXS;
  SIZE word unsigned;                  /* LENGTH OF INDEX IN BYTES
  SANITY byte unsigned;               /* SANITY CHECK (MUST = MIDXS_C_SANITY)
  constant SANITY equals 123 prefix MIDX tag $C;           /* UNUSED, ROUND SO BYTES LEFT DIV. BY 8
  FILL 1 byte dimension 5 fill prefix MIDXDEF tag $$;      /* LENGTH OF FIXED BUCKET OVERHEAD
  constant "LENGTH" equals . prefix MIDXS tag K;           /* LENGTH OF FIXED BUCKET OVERHEAD
  constant "LENGTH" equals . prefix MIDXS tag C;           /* START OF INDEX ENTRIES
  constant ENTRIES equals . prefix MIDXS tag K;            /* START OF INDEX ENTRIES
  constant ENTRIES equals . prefix MIDXS tag C;            /* EACH IS 8 BYTES,
                                                               /* 0-3: MESSAGE CODE
                                                               /* 4-7: OFFSET TO MESSAGE RECORD
                                                               /* (IF LOW BIT SET, THEN OFFSET
                                                               /* POINTS TO SUBINDEX RATHER THAN
                                                               /* A MESSAGE RECORD)

end MIDXDEF;

end_module $MIDXDEF;

module $MRECDEF;

/*
/* Message definition record in message section
*/

```

```
/*
/* All message definition records are word aligned and
/* the size is rounded to the next word boundary. This
/* is done for additional verification checks.
```

```
aggregate MRECDEF structure prefix MRECS;
  SIZE word unsigned;                                /* LENGTH OF MESSAGE DEFINITION RECORD
  TYPE byte unsigned;                               /* TYPE OF MESSAGE RECORD
  FLAGS byte unsigned;                             /* FLAGS
  LEVEL byte unsigned;                            /* MESSAGE DETAIL LEVEL (0-255)
  FAOCNT byte unsigned;                           /* NUMBER OF FAO ARGUMENTS IN MESSAGE
  USERVAL byte unsigned;                          /* USER SPECIFIED VALUE
  FILL_1 byte fill prefix MRECDEF tag $$;        /* UNUSED
  LANG byte unsigned;                            /* LANGUAGE NUMBER
  constant(
    ENGLISH                                     /* ENGLISH LANGUAGE (DEFAULT)
    , GERMAN                                     /* GERMAN LANGUAGE
    , FRENCH                                     /* FRENCH LANGUAGE
  ) equals 0 increment 1 prefix MREC tag $C;
  constant FIXEDLEN equals . prefix MRECS tag K; /* LENGTH OF FIXED PORTION OF RECORD
  constant FIXEDLEN equals . prefix MRECS tag C; /* LENGTH OF FIXED PORTION OF RECORD
  IDENTLEN byte unsigned;                        /* MESSAGE IDENT STRING LENGTH
  "IDENT" character;                            /* MESSAGE IDENT STRING
                                                /* MESSAGE TEXT FOLLOWS (ASCII)
end MRECDEF;
end_module SMRECDEF;
module SMFACDEF;
```

```
/*
/* Define facility name table within section
/*
```

```
aggregate MFACDEF structure prefix MFACS;
  NUMBER word unsigned;                           /* FACILITY NUMBER
  NAMELEN byte unsigned;                         /* LENGTH OF FACILITY NAME
  NAME character;                               /* FACILITY NAME (VARIABLE LENGTH)
end MFACDEF;
end_module SMFACDEF;
```

0251 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

MSGINT
MDL

SDLGENREQ
REQ

MSGDEF
SOL

CUTMSGCOM
FOR

CUTMSG
LIS

MSG
REQ

MAIN
LIS

LISTING
LIS